A packet routing strategy using neural networks on scale-free networks

AUTHOR(S):

Naganuma, Yuki; Igarashi, Akito

CITATION:

Naganuma, Yuki ...[et al]. A packet routing strategy using neural networks on scale-free networks. Physica A: Statistical Mechanics and its Applications 2010, 389(3): 623-628

ISSUE DATE:

2010-02

URL:

http://hdl.handle.net/2433/87417

A packet routing strategy using neural networks on scale-free networks

Yuki Naganuma, Akito Igarashi

Department of Applied Mathematics and Physics, Kyoto University, Kyoto 606-8501 Japan

Abstract

We propose a dynamic packet routing strategy by using neural networks on scale-free networks. In this strategy, in order to determine the nodes to which the packets should be transmitted, we use path lengths to the destinations of the packets, and adjust the connection weights of the neural networks attached to the nodes from local information and the path lengths. The performances of this strategy on scale-free networks which have the same degree distribution and different degree correlations are compared to one another. Our numerical simulations confirm that this routing strategy is more effective than the shortest path based strategy on scale-free networks with any degree correlations and that the performance of our strategy on assortative scale-free networks is better than that on disassortative and uncorrelated scale-free networks.

Key words: Neural network, Computer network, Packet routing, Complex network, Scale-free network
PACS: 84.35.+i, 89.75.-k, 07.05.Tp, 89.20.Hh

1 Introduction

Since the works on small-world networks by Watts and Strogatz [1] and scale-free networks by Barabási and Albert [2], the dynamics and structures of networks with the above properties have attracted a lot of interest from many fields. These networks have been called *complex networks*. Most social networks are small-world networks, in which the average shortest path length is $O(\log N)$ (N is the total number of nodes) and the neighboring nodes of a node are very often connected to each other (*high-clustering*). The world Wide Web, metabolic networks, and many social networks are scale-free networks, in which the degree distribution $P(k)$, the probability that a node is directly connected to k other nodes, is characterized by a *power law*,

$P(k) \sim k^{-\gamma}(2 < \gamma < 4)$. Besides these characteristics, various properties of networks have been researched[3; 4].

On the other hand, these days, computer networks such as the Internet have a large scale and a lot of data is communicated through them between many users. The efficiency of communication methods on the networks, therefore, has been quite important for information transfer. As a communication protocol in the Internet, we use TCP/IP, where the original data is divided into packets, which are sent to a destination of the original data through the network. There have been many previous studies on the way to transmit packets through networks [5; 6]. In these studies, it is a common assumption that the networks have a homogeneous structure, and the path to a destination is static (the shortest path is usually chosen). Faloutsos *et al.*, however, report that the Internet backbone network has a scale-free property [7], where there are nodes which are connected to a great many nodes, called *hubs*. Although hubs shorten the distance between nodes, they sometimes cause traffic congestion, since a lot of the shortest paths between nodes pass through the hubs. For this reason, efficient packet routing strategies suitable for the structures and properties of networks have been studied [8]. Many researchers have also proposed dynamic routing strategies in which information about neighbors' queue lengths is used [9; 10; 11]. Besides this, there have been studies on an optimal network structure for a routing strategy [12; 13] and a search for a path to a destination of a packet by using local information [14; 15].

As one of these strategies, Horiguchi *et al.* have proposed a routing strategy by using neural networks [16]. Their strategy uses information about queue lengths, states of neighbors and the shortest path length to the destination of a packet at each node. The assumption that each node always knows the queue lengths and states of all neighbors in every step, however, increases the information traffic in the network considerably. In order to avoid this, we propose a new strategy by using neural networks. In this strategy, only path lengths to the destinations of packets are used and we expect to avoid the congestion by adjusting the connection weights of the neural networks with local packet-discard information and path lengths. To measure the performance of our routing strategy, we run numerical simulations, where we consider a computer network as a scale-free undirected graph (network) with nodes and links for various degree correlations. Each node, which is a computer or router, has a First-In-First-Out(FIFO) queue, and generates and transfers packets. A FIFO queue means that the packet that enters a node first gets out first from the node. With the use of the routing strategy, the top packet at a node can be transferred only to a neighboring node which directly connects to the node by a communication line (a link).

In this paper, we use a routing strategy based on neural networks, where path lengths and local information are considered. In Section 2, a routing strat-

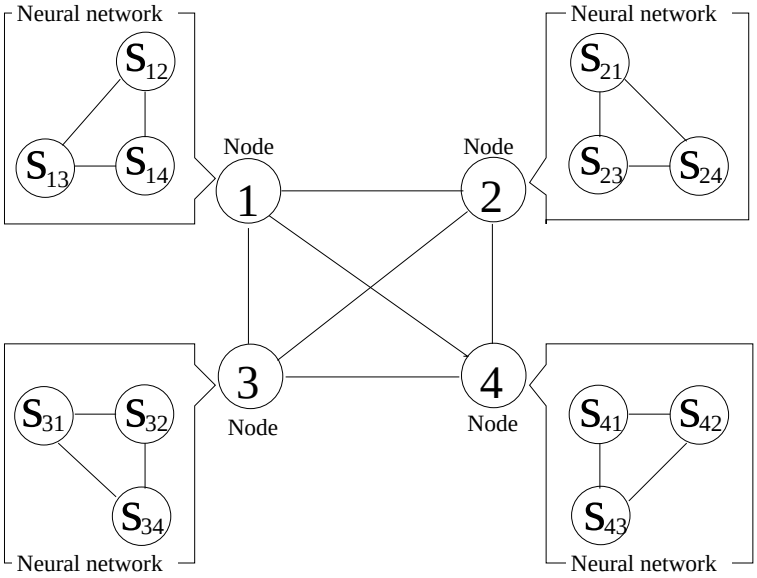

Fig. 1. A computer network and the neural networks attached to nodes.

egy using neural networks is explained and the way to adjust the connection weights of the neural networks with local information and path lengths is proposed. In Section 3, we give the results of numerical simulations on scale-free networks using our routing strategy. Finally, the conclusion is given in Section 4.

2 Model

We first define a routing strategy based on neural networks [16]. Each node has a neural network which consists of neurons corresponding to links of the node. All neurons in the neural network are connected to each other. For example, we consider a network as in Fig. 1. Since node 1 is connected to nodes 2, 3 and 4, node 1 has a neural network which consists of three neurons, s_{12}, s_{13}, and s_{14}, connected to each other.

In these neural networks, we introduce the following dynamics for each neuron.

$$v_{il}(t) = \frac{1}{1 + \exp\left[-\beta h_{il}(t)\right]} \tag{1}$$

$$h_{il}(t+1) = h_{il}(t) - \sum_{k \in \Omega(i)} J_{ik,il}(\tau) v_{ik}(t) + \left(1 - \frac{d_{l \to target(i)}}{d_i}\right) \tag{2}$$

Here $h_{il}(t)$ is the inner state of the neuron s_{il} and $v_{il}(t)$ is the output of the neuron s_{il} at iteration step t. In Eq. (1), the relation between $h_{il}(t)$ and $v_{il}(t)$

is characterized by a sigmoid function, and β is a control parameter of the sigmoid function. In Eq. (2), $\Omega(i)$ denotes the set of nodes which are directly connected to node i. $target(i)$ is the destination of the packet which is at the top of the queue of node i, and $d_{l\to t}$ the shortest path length from node l to node t. d_i is defined as $d_i = \max_t d_{i\to t}$. $J_{ik,il}(\tau)$ stands for the connection weight between neuron s_{ik} and neuron s_{il} at transmission time τ. Every transmission time, each node sends a packet to the next node if the node has at least one packet. The way to decide the next target node of the top packet at each node is as follows. First, $h_{il}(0)$, $v_{il}(0)$ and $d_{l\to target(i)}$ for $\forall i$ and $\forall l$ are initialized. Second, by using Eqs. (1) and (2) at each neuron in each node, the inner state and the output of the neuron are updated simultaneously and $t \leftarrow t+1$. After F iterations of this update, that is at $t = F$, the node where the packet at the top of the queue in node i is transmitted is provided by $\arg\max_{l\in\Omega(i)} v_{il}(F)$, and we increase the transmission time, that is, $\tau \leftarrow \tau + 1$. This process is iterated.

Moreover, we propose two adjustive rules for the connection weights, $J_{ik,il}(\tau)$: the *discard adjustment* using local packet-discard information and the *direction adjustment* using path lengths to the destinations of the packets.

The *discard adjustment* is as follows: we adjust (increase) the connection weights $J_{ik,il}$ in order to avoid transmitting a packet from node i to neighbors l and k if packets transmitted from node i have been frequently discarded at the neighbors l and k. For evaluation of the packet-discard frequency, we introduce the *discard weight* $L_{il}(\tau)$ at transmission time τ, which is the number of packets which have been discarded until transmission time $\tau - 1$ for the transmission from node i to neighbor $l \in \Omega(i)$, and is renewed as $L_{il}(\tau+1) = L_{il}(\tau) + e_{il}(\tau)$, where $e_{il}(\tau)$, expressing whether a packet transmitted from node i to a neighbor l is discarded at transmission time τ, is set to be 1 (if a packet is transmitted from node i to node l and discarded at node l) or 0 (otherwise). At each transmission time τ, we define $\Lambda_i^L(\tau)$ as the set of indexes of $g_i^L(\tau)$ neighbors at each node i, $\{l_1, l_2, \cdots, l_{g_i^L(\tau)}\}$, which satisfies the following inequality: $L_{i,l_1}(\tau) \geq L_{i,l_2}(\tau) \geq \cdots \geq L_{i,l_{g_i^L(\tau)}}(\tau) \geq L_{i,l_{g_i^L(\tau)+1}}(\tau) \geq \cdots \geq L_{i,l_{k_i}}(\tau)$, where k_i is the degree of node i. That is, we arrange $\{L_{il}(\tau)\}$ in descending order for fixed i and choose $g_i^L(\tau)$ indexes from the index with the largest discard weight to that with the $g_i^L(\tau)$th value. We update the connection weight, $J_{ik,il}$ as follows: for $k, l \in \Omega(i)$ and $k \neq l$,

$$J_{ik,il}(\tau+1) = \begin{cases} J_L + \delta J_L & (k \in \Lambda_i^L(\tau),\ l \in \Lambda_i^L(\tau)\ ) \\ J_L - \delta J_L & (k \notin \Lambda_i^L(\tau),\ l \notin \Lambda_i^L(\tau)\ )\ . \\ J_L & (\text{otherwise}\ ) \end{cases} \tag{3}$$

Here J_L and $\delta J_L (> 0)$ are parameters. In this adjustment, the discard in-

formation which we use is local, since it is informed to a node only from a nearest node when a packet is discarded at the node when it is transmitted. Because discarding of a packet does not always occur at a neighbor in every transmission time, the traffic is not so enhanced with this local discard information.

The *direction adjustment* is as follows. At each transmission time τ, we divide the neighbors at each node into two groups. One group, $\Lambda_i^D(\tau)$, is the set of neighbors which are near the destination of the top packet at node i. The other group is the complement set of $\Lambda_i^D(\tau)$. That is to say, at each transmission time τ, we choose the $g_i^D(\tau)$ neighbors selected by starting from the smallest value to the $g_i^D(\tau)$th value of the shortest path lengths from the neighbor nodes to the destination of the top packet at node i. We define $\Lambda_i^D(\tau)$ as the set of those $g_i^D(\tau)$ neighbors. By using this path length information, the connection weights are adjusted and the packets are flexibly transmitted to one of the neighboring nodes near the destination. We update the connection weights $J_{ik,il}$ as follows: for $k, l \in \Omega(i)$ and $k \neq l$,

$$J_{ik,il}(\tau+1) = \begin{cases} J_D + \delta J_1 & (k \in \Lambda_i^D(\tau),\ l \in \Lambda_i^D(\tau)\ ) \\ J_D + \delta J_2 & (k \notin \Lambda_i^D(\tau),\ l \in \Lambda_i^D(\tau)\ ) \\ J_D + \delta J_3 & (k \in \Lambda_i^D(\tau),\ l \notin \Lambda_i^D(\tau)\ ) \\ J_D + \delta J_4 & (k \notin \Lambda_i^D(\tau),\ l \notin \Lambda_i^D(\tau)\ ) \end{cases}. \tag{4}$$

Here, $J_D,\ \delta J_1(<0),\ \delta J_2,\ \delta J_3$, and $\delta J_4(>0)$ are parameters.

In the next section, we will give some results obtained by means of numerical simulations by using neural networks and these adjustive rules on some scale-free networks.

3 Simulations

We run numerical simulations under the following assumptions. Each node has a First-In-First-Out(FIFO) queue and the maximum length of the queue, called the buffer size, is finite. The buffer size at node l, b_l, is 20 for $\forall\ l$ in this paper. A packet is removed from a network if the packet arrives at its destination (called *arrival*) or is transmitted to a next node of which the queue is full (called *discarding*). The number of packets in the network, N_p, is always fixed since a new packet is generated at a randomly selected node whose queue is not full and is queued at the end if a packet arrives at the destination or is discarded. The source and destination of the generated packet are determined

at random and each node has enough information to determine the next node of the top packet. Each link is a full duplex; that is, each link is undirected. The time interval to transmit a packet between nodes is constant ($F = 50$), and all nodes send packets simultaneously.

We define the average queue length, $\langle q \rangle$, the average number of packets that arrive at their destinations per transmission time, A, and the average number of packets discarded without arriving at their destinations per transmission time, D, as follows:

$$\langle q \rangle = \frac{N_p}{N}, \quad A = \frac{N_a}{N \times T}, \quad D = \frac{N_d}{N \times T} \tag{5}$$

Here N is the total number of nodes in the network and T the final value of the transmission time of one numerical simulation. N_p denotes the total number of packets in the network, N_a the number of packets which have arrived at their destinations during T and N_d the number of packets which have been discarded without arriving at their destinations during T. In the present paper, we set $T = 300$.

We use the *direction adjustment* after the *discard adjustment* as an adjustive rule of neural networks. The *direction adjustment* after the *discard adjustment* means that at transmission time τ the *discard adjustment* is performed for $J_L = J_{ik,il}(\tau)$ and after that, the *direction adjustment* is performed with J_D as the value of $J_{ik,il}$ determined by the *discard adjustment*. For comparison, we introduce the shortest path (SP) based strategy which means that each node sends a packet toward the neighbor where the path length to its destination is the smallest.

First, we use the Barabási and Albert(BA) model [2] as the network topology. This model is constructed as follows. First, we start from a network which has some nodes and links. Second, we add a new node with m links (*growth*) and neighboring nodes of the new node are selected and linked in probability proportional to their degrees of the nodes in the network (*preferential attachment*). By repeating *growth* and *preferential attachment*, we obtain a scale-free network which has the degree distribution $P(k) \sim k^{-3}$ and an average degree $\langle k \rangle = 2m$.

In Figs. 2 and 3, we show numerical results on the BA model, comparing these strategies. Here, we set $\beta = 3$, $J_{ik,il}(\tau = 0) = 1$ for $\forall\, i, \forall\, k$ and $\forall\, l$, $\delta J_L = 0.2$, $\delta J_1 = -0.15$, $\delta J_4 = 0.15$, and $\delta J_2 = \delta J_3 = 0$. $g_i^L(\tau)$ and $g_i^D(\tau)$ are set to be the integer part of $k_i/2$, where k_i is the degree of node i. From these figures, it is found that the performance of our strategy is better than that of the SP based strategy, that is; A of our strategy is larger than that of the SP based strategy and D of our strategy is smaller. These results mean that congested

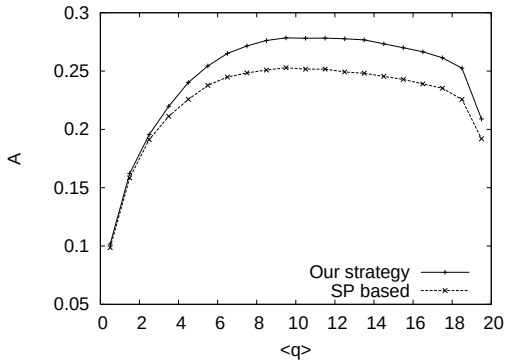

Fig. 2. The average number of packets arriving at their destinations per transmission time, A, vs. $\langle q \rangle$. Results with our strategy and the SP based strategy are compared on the BA model for $N = 100$ and $\langle k \rangle = 10$.

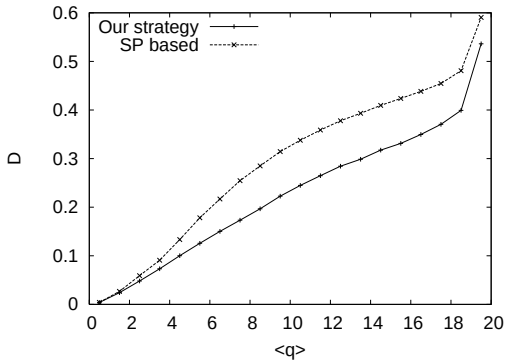

Fig. 3. The average number of packets discarded per transmission time, D, vs. $\langle q \rangle$. Results with our strategy and the SP based strategy are compared on the BA model for $N = 100$ and $\langle k \rangle = 10$.

neighbor nodes can be efficiently avoided by using the adjustment of neural networks.

Second, we run numerical simulations on scale-free networks which have different degree correlations from one another. The degree correlation of a network means the average correlation between the degrees of two end nodes of a link in the network, and the degree correlation coefficient, r_d, is defined as follows:

$$r_d = \frac{\sum_n^M \frac{\kappa_n \kappa'_n}{M} - \left(\frac{\sum_n^M \kappa_n}{M}\right)\left(\frac{\sum_n^M \kappa'_n}{M}\right)}{\sqrt{\frac{\sum_n^M \kappa_n^2}{M} - \left(\frac{\sum_n^M \kappa_n}{M}\right)^2}\sqrt{\frac{\sum_n^M {\kappa'_n}^2}{M} - \left(\frac{\sum_n^M \kappa'_n}{M}\right)^2}}. \tag{6}$$

Here M is the total number of links in the network, and κ_n and κ'_n are the *excess degrees* of the nodes at the two ends of the nth link, respectively. The *excess degree* is one less than the degree of a node. A network for $r_d > 0$ is called an assortative network, and a network for $r_d < 0$ is called a disassortative network. In order to tune the degree correlation of the network, we use the rewiring algorithm which is proposed by Newman [17]. In this algorithm, we obtain a network of an arbitrary degree correlation by rewiring links of a network which has been originally generated (the BA model, which is uncorrelated, $r_d = 0$, is used in this paper). This algorithm is as follows. We choose two links (u_1, w_1) and (u_2, w_2) at random from the network. We measure the *excess degrees* $\kappa_1, \kappa'_1, \kappa_2$ and κ'_2 of the nodes u_1, w_1, u_2 and w_2. We remove the two links and replace them with two new ones (u_1, u_2) and (w_1, w_2) with probability $P_c = e_{\kappa_1\kappa_2} e_{\kappa'_1\kappa'_2} / e_{\kappa_1\kappa'_1} e_{\kappa_2\kappa'_2}$ (if $e_{\kappa_1\kappa_2} e_{\kappa'_1\kappa'_2} < e_{\kappa_1\kappa'_1} e_{\kappa_2\kappa'_2}$) or 1 (otherwise). We choose $e_{\kappa\kappa'}$ as $e_{\kappa\kappa'} = 2q(\kappa)q(\kappa') - q(\kappa)x(\kappa') - x(\kappa)q(\kappa') + x(\kappa)x(\kappa')$ (to generate an assortative network) or $e_{\kappa\kappa'} = q(\kappa)x(\kappa') + x(\kappa)q(\kappa') - x(\kappa)x(\kappa')$ (to

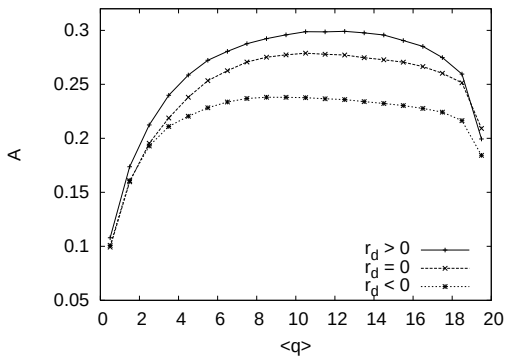

Fig. 4. The average number of packets arriving at their destinations per transmission time, A, vs. $\langle q \rangle$. Results with our strategy for assortative, uncorrelated and disassortative degree correlations are compared for $N = 100$ and $\langle k \rangle = 10$.

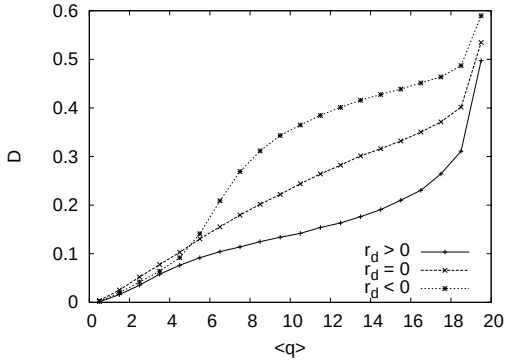

Fig. 5. The average number of packets discarded per transmission time, D, vs. $\langle q \rangle$. Results with our strategy for assortative, uncorrelated and disassortative degree correlations are compared for $N = 100$ and $\langle k \rangle = 10$.

generate a disassortative network). Here $q(\kappa)$ stands for the distribution of the excess degree of a node at an end of a link, given by $(\kappa+1)P(\kappa+1)/\sum_k kP(k)$, where $P(k)$ is the degree distribution. $P(k)$ is unchanged in the rewiring process, and $x(\kappa)$ is an arbitrary distribution function which reduces more rapidly, as κ increases, than $q(\kappa)$. By repeating this rewiring sufficiently, we obtain a network with an arbitrary degree correlation if a suitable $x(\kappa)$ is used. In this paper, we set $x(\kappa)$ as $x(\kappa) \sim \mathrm{e}^{-\kappa} q(\kappa)$.

Our strategy performs both in assortative and disassortative networks more efficiently than the SP based strategy, as in the uncorrelated (BA) network shown in Figs. 2 and 3. In Figs. 4 and 5, we show numerical results obtained by the strategy using neural networks on scale-free networks which have three different degree correlations from one another ($r_d = 0.25, 0, -0.5$). By using the SP strategy, similar but less efficient results are obtained.

From these figures, it is shown that, as r_d increases, that is, the network becomes assortative, A increases and D decreases for $\langle q \rangle \gtrsim 6$. This is because, in a disassortative network, there are many low-degree nodes linked to two high-degree nodes and a path to avoid high-degree nodes does not often exist, so congestion occurs at high-degree nodes with high probability and A becomes small. On the other hand, in an assortative network, there are some clusters where high-degree nodes connect to each other and another path to avoid congested high-degree nodes usually exists, so A becomes large. From this result, even if networks have the same average degree and degree distribution, the routing efficiency depends on its degree correlation and the performance of our strategy on assortative scale-free networks is more efficient than that on disassortative and uncorrelated scale-free networks.

4 Conclusion

In the present paper, we propose a routing strategy by using neural networks and adjustive rules of the connection weights. The adjustive rules are performed using local information such as whether a packet is discarded at one of the nearest nodes and the path length to the destination of the top packet of the queue at each node. First, we confirm that the performance of our routing strategy is more efficient than the performance of the SP based strategy on scale-free networks. Moreover, we run numerical simulations on scale-free networks with various degree correlations. From these simulations, it is found that the performance of this routing strategy is enhanced for any degree correlations compared with that of the SP based strategy, and is improved as the degree correlation coefficient of the scale-free network becomes large. This is because high-degree nodes tend to connect to one another in high degree correlation (assortative) scale-free networks.

As for future problems, we hope to devise some adjustive rules other than the rules proposed in this paper, with which the performance of the routing strategy will improve.

References

[1] D. J. Watts, S. H. Strogatz, Nature 286 (1998) 440.
[2] A. L. Barabási, R. Albert, Science 286 (1999) 509.
[3] M. E. J. Newman, SIAM Review 45 (2003) 167.
[4] S. Boccaleti, V. Latora, Y. Moreno, M. Chavez and D. U. Hwang, Phys. Rep. 424 (2006) 175.
[5] R. Bellman, Quarterly of Applied Math. 16 (1959) 87.
[6] D. Bertsekas, R. Gallager, Data Networks, Prentice-Hall, 1987.
[7] M. Faloutsos, P. Faloutsos, C. Faloutsos, ACM SIGCOM Computer Com. Rev. 29 (1999) 251.
[8] G. Yan, T. Zhou, B. Hu, B, Wang, Phys. Rev. E 73 (2006) 046108.
[9] Z. Y. Chen, X. F. Wang, Phys. Rev. E 73(2006) 036107.
[10] P. Echenique, J. Gómez-Gardeñes, Y. Moreno, Phys. Rev. E 70 (2004) 056105.
[11] P. Echenique, J. Gómez-Gardeñes, Y. Moreno, Euro. Phys. Lett. 71 (2005) 325.
[12] R. Guimerá, A. Díaz-Guilera, F. Vega-Redondo, A. Cabrales, A. Arenas, Phys. Rev. Lett. 89 (2002) 248701.
[13] D. J. Watts, P. S. Dodds, M. E. J. Newman, Science 296 (2002) 1302.
[14] B. Danila, Y. Yu, J. A. Marsh, K. E. Bassler, Phys. Rev. E 74 (2006) 046106.

[15] S. Sreenivasan, R. Cohen, E. López, Z. Toroczkai, H. E. Stanley, Phys. Rev. E 75 (2007) 036105.
[16] T. Horiguchi, S. Ishioka, Physica A 297 (2001) 521.
[17] M. E. J. Newman, Phys. Rev. E 67 (2003) 026126.